

Circuit Bottom Fan-in and Computational Power

Liming Cai
School of Electrical and Computer Science
Ohio University
Athens, Ohio 45701, USA
cai@prime.cs.ohiou.edu

Jianer Chen
Department of Computer Science
Texas A&M University
College Station, TX 77843, USA
chen@cs.tamu.edu

Johan Håstad
Department of Computer Science
Royal Institute of Technology
Stockholm, Sweden
johanh@nada.kth.se

Abstract

We investigate the relationship between circuit bottom fan-in and circuit size when circuit depth is fixed. We show that in order to compute certain functions, a moderate reduction in circuit bottom fan-in will cause significant increase in circuit size. In particular, We prove that there are functions that are computable by circuits of linear size and depth k with bottom fan-in 2 but require exponential size for circuits of depth k with bottom fan-in 1. A general scheme is established to study the trade-off between circuit bottom fan-in and circuit size. Based on this scheme, we are able to prove, for example, that for any integer c , there are functions that are computable by circuits of linear size and depth k with bottom fan-in $O(\log n)$ but require exponential size for circuits of depth k with bottom fan-in c , and that for any constant $\epsilon > 0$, there are functions that are computable by circuits of linear size and depth k with bottom fan-in $\log n$ but require superpolynomial size for circuits of depth k with bottom fan-in $O(\log^{1-\epsilon} n)$. A consequence of these results is that the three input read-modes of alternating Turing machines proposed in the literature are all distinct.

1 Introduction

To prove lower bounds for various computational models remains as one of the most challenging task in complexity theory. Much progress has been made recently in deriving lower bounds for computational models with limited capabilities, with the hope that these may lead to better lower bounds for more general computational models and to better

understanding of intrinsic complexity of computation.

One of the most successful trials is the derivation of lower bounds for constant depth circuits. The first strong lower bounds were given by Furst, Saxe, and Sipser [11], independently by Ajtai [1], who show that the size of a constant depth circuit computing parity function is superpolynomial. The results were subsequently sharpened by Yao [17] who derived an exponential lower bound. Håstad [13, 14] further strengthened the result and obtained near optimal lower bounds. A direct consequence of these results is that the logarithmic time hierarchy [16], i.e., the set of languages accepted by families of circuits of constant depth and polynomial size, is a proper subset of P .

The logarithmic time hierarchy was further refined by Sipser [16] who showed that for each integer $k > 1$, there are functions that are computable by a circuit of depth k and polynomial size but require superpolynomial size for circuits of depth $k - 1$. Thus, all levels of the logarithmic time hierarchy are distinct. Exponential lower bounds for the depth k to $k - 1$ conversion were sequentially claimed by Yao [17] and fully proved by Håstad [13, 14].

In this paper, we will further sharpen the separation results in the logarithmic time hierarchy by investigating the relationship between circuit bottom fan-in and circuit size when circuit depth is fixed. We show that in order to compute certain functions, a moderate reduction in circuit bottom fan-in will cause significant increase in circuit size. In particular, We prove that there are functions that are computable by circuits of linear size and depth k with bottom fan-in 2 but require exponential size for circuits of depth k with bottom fan-in 1. A general scheme is established to study the trade-off between circuit bottom fan-in and circuit size. Based on this scheme, we are able to prove, for

example, that for any integer c , there are functions that are computable by circuits of linear size and depth k with bottom fan-in $O(\log n)$ but require exponential size for circuits of depth k with bottom fan-in c , and that for any constant $\epsilon > 0$, there are functions that are computable by circuits of linear size and depth k with bottom fan-in $\log n$ but require superpolynomial size for circuits of depth k with bottom fan-in $O(\log^{1-\epsilon} n)$. Therefore, the computational power of constant depth circuits depends not only on its depth, but also strictly on its bottom fan-in when the depth of the circuits is fixed.

Another motivation of our present research is from the study of input read-modes of a sublinear-time alternating Turing machine, which is an important computational model in the study of complexity classes. A number of input read-modes for sublinear-time alternating Turing machines have appeared in the literature. In the standard model proposed by Chandra, Kozen, and Stockmeyer [8], a computation path of the machine can read up to $O(\log n)$ input bits in time $O(\log n)$. Ruzzo [15] proposed an input read-mode in which each computation path can read at most one input bit and the reading must be performed at the end of the path. An input read-mode studied by Sipser [16] insists that each input reading takes time $\Omega(\log n)$. These input read-modes have been carefully studied by Cai and Chen [5], who have given a precise circuit characterization for each read-mode for log-time alternating Turing machines of constant alternations. Input read-modes of log-time alternating Turing machines also find applications in the study of computational optimization problems [6, 7].

Based on Cai and Chen's circuit characterizations and our separation results in constant depth circuits, we are able to show that the three proposed input read-modes for alternating Turing machines are all distinct. More precisely, if we let Π_k^U (resp. Π_k^R, Π_k^S) be the class of languages accepted by log-time k -alternation alternating Turing machines using Chandra, Kozen, and Stockmeyer's (resp. Ruzzo's, Sipser's) input read-mode, then we can show that for all integers $k \geq 1$

$$\Pi_k^R \subset \Pi_k^S \subset \Pi_k^U \subset \Pi_{k+1}^R$$

where \subset means "proper subset". This gives a very detailed refinement of the logarithmic time hierarchy, and shows the rich structural properties of the logarithmic time hierarchy.

We briefly review the fundamentals related to the present paper. For further discussion on the theory of circuit complexity and alternating Turing machines, the reader is referred to [3, 10].

An (unbounded fan-in) *Boolean circuit* α_n with input $x = x_1 x_2 \cdots x_n$ of length n is a directed acyclic graph. The *fan-in* of a node in the circuit is the in-degree of the node. The nodes of fan-in 0 are called *inputs* and are labeled from the set $\{0, 1, x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$. The nodes of fan-in greater than 0 are called *gates* and are labeled either

AND or OR. One of the nodes is designated the *output* node. The *size* is the number of gates, and the *depth* is the maximum distance from an input to the output. Without loss of generality, we assume the circuits are of the special form where all AND and OR gates are organized into alternating levels with edges only between adjacent levels. Any circuit may be converted to one of this form without increasing the depth and by at most squaring the size [9]. In this special form, the gates that are connected to input nodes will be called *bottom level gates*, or *depth 1 gates*. The gates that receive inputs from depth 1 gates will be called *depth 2 gates*, and so on. The *bottom fan-in* of a circuit is the maximum over fan-ins of all bottom level gates. The following notations introduced by Boppana and Sipser [3] will be especially convenient in our discussion.

Definition [3] A circuit α is a Π_k^s -circuit (resp. Σ_k^s -circuit) if α is a depth k circuit of size at most s with an AND-gate (resp. an OR-gate) at the output. A circuit β is a $\Pi_k^{s,c}$ -circuit (resp. $\Sigma_k^{s,c}$ -circuit) if β is a depth $k+1$ circuit of size at most s with bottom fan-in c and an AND-gate (resp. an OR-gate) at the output.

A *family* of circuits is a sequence $\{\alpha_n \mid n \geq 1\}$ of circuits, where α_n is with input of length n . A family of circuits may be used to define a language. A family $\{\alpha_n \mid n \geq 1\}$ of circuits is said to be a Π_k^{poly} -family (resp. $\Pi_k^{poly,c}$ -family) if there is a polynomial p such that for all $n \geq 1$, α_n is a $\Pi_k^{p(n)}$ -circuit (resp. $\Pi_k^{p(n),c}$ -circuit).

The *Sipser function* f_k^m , as defined in [13, 14], is given by a tree circuit of depth k , in which every gate in the bottom level has fan-in $\sqrt{km \log m/2}$, the fan-in of the output gate is $\sqrt{m/\log m}$, and the fan-in for all other gates is m . Each variable x_i , $1 \leq i \leq n$, occurs at only one leaf. Note that the number n of variables of the function f_k^m equals $m^{k-1} \sqrt{k/2}$. The following theorem is proved by Håstad [13, 14].

Theorem 1.1 ([13, 14]) *There is no depth k circuit computing the Sipser function f_k^m with bottom fan-in $\frac{1}{12\sqrt{2k}} \sqrt{\frac{m}{\log m}}$ and less than $2^{\frac{1}{12\sqrt{2k}} \sqrt{\frac{m}{\log m}}}$ gates of depth ≥ 2 , for $m > m_0$, where m_0 is a absolute constant.*

The discussion of the present paper is centered on the complexity of the function $f_k^{m,b}$, formally defined as follows.

Definition Let $C_k^{m,b}$ be the tree circuit defining the Sipser function f_k^m , except that each bottom level gate of $C_k^{m,b}$ has fan-in b instead of $\sqrt{km \log m/2}$. Define $f_k^{m,b}$ to be the function computed by the tree circuit $C_k^{m,b}$.

An $O(\log n)$ -time alternating Turing machine (log-time ATM) is defined as an extension of the $O(\log n)$ -time deterministic Turing machines in the usual way [8]. Given an input, the computation of a log-time ATM M can be represented by an \wedge - \vee tree. Each computation path in the \wedge - \vee tree can be divided into *phases*, which are the maximal subpaths in which M does not make alternations. The first configuration in each phase is called an *alternation* (configuration). In particular, the starting configuration of M is always an alternation.

2 On circuits that compute $f_k^{m, \Omega(\log m)}$

In this section, we consider the complexity of the function $f_k^{m, b}$, where $b = \Omega(\log m)$. Note that the number n of variables of the function $f_k^{m, b}$ is $n = bm^{k-2} \sqrt{m/\log m}$.

We first consider the case $b \leq \sqrt{km \log m/2}$.

Theorem 2.1 For $k \log m < b \leq \sqrt{km \log m/2}$, the function $f_k^{m, b}$ cannot be computed by any depth k circuit of bottom fan-in $\frac{b}{12k \log m}$ and size bounded by $2^{\frac{1}{12\sqrt{2(k-1)}} \sqrt{\frac{m}{\log m}}}$, for $m > m_0$, where m_0 is an absolute constant.

PROOF (SKETCH). We set $q = \frac{k \log m}{b}$. Let $(B_j)_{j=1}^r$ be the partition of the variables of the function $f_k^{m, b}$ such that each block B_j is the set of variables leading into the same bottom level gate of the tree circuit $C_k^{m, b}$. Let $R_{q, B}^+$, $R_{q, B}^-$, and $\rho g(\rho)$ be the probability spaces and the restriction introduced by Håstad [13], respectively.

Suppose that the theorem is not true. Thus, there is a depth k circuit C_0 of size bounded by $2^{\frac{1}{12\sqrt{2(k-1)}} \sqrt{\frac{m}{\log m}}}$ and bottom fan-in $t \leq \frac{b}{12k \log m}$ that computes the function $f_k^{m, b}$. Then with the probability spaces $R_{q, B}^+$ and $R_{q, B}^-$ and the restriction $\rho g(\rho)$, and using the switching lemma [13], we are able to construct a circuit that contradicts Theorem 1.1 (The proof is similar to the induction step in the proof given by Håstad for Theorem 1.1). \square

Let $b = \sqrt{km \log m/2}$ in Theorem 2.1, we obtain Theorem 1.1. Note that the size bound is slightly improved.

The condition $b \leq \sqrt{km \log m/2}$ in Theorem 2.1 is essential in its proof. For larger bottom fan-in b , we have the following theorem.

Theorem 2.2 For $b \geq 2\sqrt{km \log m/2}$, the function $f_k^{m, b}$ cannot be computed by any depth k circuit of bottom fan-in $\frac{b}{25k e \log m}$ and size bounded by $2^{\frac{1}{12\sqrt{2k}} \sqrt{\frac{m}{\log m}}}$, for $m > m_0$, where m_0 is an absolute constant and e is the base of the natural logarithm.

PROOF. Let $q = \frac{1.04 \sqrt{km \log m/2}}{b}$. Consider the following probability space R_q^+ of restrictions:

For each variable x_j of the function $f_k^{m, b}$, let $\rho^+(x_j) = *$ with probability q and else $\rho^+(x_j) = 1$.

The probability space R_q^- is defined similarly except that the value 1 is replaced by value 0.

From now on, we assume that the bottom level gates of the tree circuit $C_k^{m, b}$ defining $f_k^{m, b}$ are AND gates. The case when the bottom level gates of $C_k^{m, b}$ are OR gates can be proved similarly by using the probability space R_q^- instead of the probability space R_q^+ .

We first show that under a restriction $\rho^+ \in R_q^+$, with very large probability, the tree circuit $C_k^{m, b}$ computes a function at least as hard as the Sipser function f_k^m .

Let τ be a bottom level gate in the tree circuit $C_k^{m, b}$. The gate τ is an AND gate of fan-in b . Let $p_i = \binom{b}{i} q^i (1-q)^{b-i}$ be the probability that the gate τ gets exactly i $*$'s under a restriction $\rho^+ \in R_q^+$. First we consider the ratio

$$\frac{p_i}{p_{i-1}} = \frac{b-i+1}{i} \cdot \frac{q}{1-q} > \frac{b-i}{i} \cdot \frac{q}{1-q}$$

For $i \leq 1.02\sqrt{km \log m/2}$, we have

$$\frac{b-i}{i} \geq \frac{b-1.02\sqrt{km \log m/2}}{1.02\sqrt{km \log m/2}}$$

and

$$\begin{aligned} \frac{q}{1-q} &= \frac{(1.04\sqrt{km \log m/2})/b}{1 - (1.04\sqrt{km \log m/2})/b} \\ &= \frac{1.04\sqrt{km \log m/2}}{b - 1.04\sqrt{km \log m/2}} \end{aligned}$$

Thus, we have

$$\begin{aligned} \frac{p_i}{p_{i-1}} &> \frac{b-1.02\sqrt{km \log m/2}}{1.02\sqrt{km \log m/2}} \cdot \frac{1.04\sqrt{km \log m/2}}{b-1.04\sqrt{km \log m/2}} \\ &\geq \frac{52}{51} \end{aligned}$$

This gives $(51/52)^{j-i} p_j > p_i$ for $i < j \leq 1.02\sqrt{km \log m/2}$.

Now under a restriction $\rho^+ \in R_q^+$, the probability that the gate τ gets less than $\sqrt{km \log m/2}$ $*$'s is bounded by

$$\begin{aligned} &\sum_{i=0}^{\sqrt{km \log m/2}} p_i \\ &< \sum_{i=0}^{\sqrt{km \log m/2}} (51/52)^{\sqrt{km \log m/2}-i} p_{\sqrt{km \log m/2}} \end{aligned}$$

$$\begin{aligned}
&\leq 52p\sqrt{km \log m/2} \\
&< 52(51/52)^{0.02\sqrt{km \log m/2}} p_{1.02\sqrt{km \log m/2}} \\
&\leq 52(51/52)^{0.02\sqrt{km \log m/2}}
\end{aligned}$$

and $52(51/52)^{0.02\sqrt{km \log m/2}}$ is smaller than $\frac{1}{m^k}$ for sufficiently large m .

Since the circuit $C_k^{m,b}$ has less than m^{k-1} bottom level gates, we conclude that under a restriction $\rho^+ \in R_q^+$, the probability that any bottom level gate of the tree circuit $C_k^{m,b}$ gets less than $\sqrt{km \log m/2}$ *'s is bounded by $\frac{1}{m}$.

Now suppose that the theorem is not true. Thus, there is a depth k circuit C_0 of bottom fan-in at most $\frac{b}{25k e \log m}$ and size bounded by $2^{\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}}$ such that the circuit C_0 computes the function $f_k^{m,b}$. We show that under a restriction $\rho^+ \in R_q^+$, with very large probability, the circuit C_0 becomes a depth k circuit of bottom fan-in at most $\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}$.

Let μ be a bottom level gate of fan-in $c \leq \frac{b}{25k e \log m}$ in the circuit C_0 , and let $r_i = \binom{c}{i} q^i (1-q)^{c-i}$ be the probability that the gate μ gets exactly i *'s under a restriction $\rho^+ \in R_q^+$. We have

$$\binom{c}{i} q^i (1-q)^{c-i} \leq \frac{c!}{i!(c-i)!} q^i \leq \frac{c^i q^i}{i!} \leq \left(\frac{cqe}{i}\right)^i$$

where the last inequality is based on Stirling's approximation [12]

$$i! \geq 0.9(i/e)^i \sqrt{2\pi i} \geq (i/e)^i, \quad \text{for } i \geq 1$$

Let $s = \frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}$. Under a restriction $\rho^+ \in R_q^+$ the probability that the gate μ gets more than s *'s is bounded by

$$\sum_{i=s+1}^c r_i \leq \sum_{i=s+1}^c \left(\frac{cqe}{i}\right)^i \leq \sum_{i=s+1}^c \left(\frac{1.04}{25\sqrt{2k}}\sqrt{\frac{m}{\log m}}\right)^i$$

For $i > s = \frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}$, we have $\left(\frac{1.04}{25\sqrt{2k}}\sqrt{\frac{m}{\log m}}\right)/i < \frac{12.48}{25}$. Thus under a restriction $\rho^+ \in R_q^+$ the probability that the gate μ gets more than s *'s is bounded by

$$\sum_{i=s+1}^c (12.48/25)^i < (12.48/25)^s$$

Since the circuit C_0 has at most 2^s bottom level gates, we conclude that under a restriction $\rho^+ \in R_q^+$, the probability that any bottom level gate of the circuit C_0 gets more than s *'s is bounded by

$$\begin{aligned}
(12.48/25)^s \cdot 2^s &= (24.96/25)^s \\
&= (24.96/25)^{\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}}
\end{aligned}$$

which is smaller than $\frac{1}{m}$ for sufficiently large m .

Thus, under a restriction $\rho^+ \in R_q^+$, with probability $\geq 1 - \frac{1}{m} - \frac{1}{m} > \frac{1}{2}$, all bottom level gates of the tree circuit $C_k^{m,b}$ get at least $\sqrt{km \log m/2}$ *'s (thus $C_k^{m,b}$ is converted to a circuit computing a function at least as hard as the Sipser function f_k^m), and all bottom level gates of the circuit C_0 get at most $\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}$ *'s. Note that if a bottom level gate μ of the circuit C_0 gets at most $\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}$ *'s, then either the gate μ is eliminated from the bottom level (e.g., μ is an AND gate and gets an input with value 0) or the gate μ becomes a gate of fan-in at most $\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}$. In any case, we have derived that there is an assignment that converts the circuit C_0 into a depth k circuit C' of bottom fan-in bounded by $\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}$ and size bounded by $2^{\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}}$ such that the circuit C' computes a function at least as hard as the Sipser function f_k^m . But this contradicts Theorem 1.1. \square

3 On circuits that compute $f_k^{m,2}$

In the previous section, we showed that for circuits to compute the function $f_k^{m,b}$, $b = \Omega(\log m)$, an $O(\log m)$ factor reduction in bottom fan-in may cause an exponential increase in the circuit size. In this section, we will show that in certain cases, even reducing the circuit bottom fan-in by 1 will cause an exponential increase in the circuit size. More precisely, we will show that the function $f_k^{m,2}$ can be computed by a depth k circuit of linear size and bottom fan-in 2, but requires exponential size for depth k circuits of bottom fan-in 1. Note that a depth k circuit of bottom fan-in 1 is actually a depth $k-1$ circuit.

We prove the above result with a new probability space of restrictions. We start with the following lemmas.

Lemma 3.1 *Partition the Boolean variables $\{x_1, \dots, x_n\}$ into groups of c variables each. For each group, randomly pick r variables and assign them 0. Let σ be an OR of a subset S_σ of $\{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$ such that S_σ contains at least h negative literals \bar{x}_i . Then with the above random assignment,*

$$Pr[\sigma \neq 1] \leq ((c-r)/c)^h$$

Lemma 3.2 *Partition the Boolean variables $\{x_1, \dots, x_n\}$ into groups of c variables each. For each group, randomly pick r variables and assign them 1. Let σ be an OR of a subset S_σ of $\{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$ such that S_σ contains at least h positive literals x_i . Then with the above random assignment,*

$$Pr[\sigma \neq 1] \leq ((c-r)/c)^h$$

Now we are ready for the main theorem of this section.

Theorem 3.3 *The function $f_k^{m,2}$ cannot be computed by any depth $k - 1$ circuit of size bounded by $2^{\frac{1}{12\sqrt{2(k-1)}}\sqrt{\frac{m}{\log m}}}$ for $m > m_0$, where m_0 is an absolute constant.*

PROOF. To simplify the expressions, we let $s = \frac{1}{12\sqrt{2(k-1)}}\sqrt{\frac{m}{\log m}}$. Suppose that the theorem is not true and that there is a depth $k - 1$ circuit C of size 2^s that computes the function $f_k^{m,2}$. Furthermore, we assume that the gates in the bottom level of C are OR gates (the case that the bottom level gates of C are AND gates can be proved similarly.)

Randomly pick one variable from each pair x_{2i-1} and x_{2i} and assign it 0. This will reduce the tree circuit $C_k^{m,2}$ defining $f_k^{m,2}$ to the tree circuit $C_{k-1}^{m,m}$ defining $f_{k-1}^{m,m}$.

Let τ be an OR gate in the bottom level of the circuit C such that τ has more than s negative literals in its input, then by Lemma 3.1,

$$Pr[\tau \neq 1] \leq (1/2)^{s+1}$$

Let $\tau_1, \dots, \tau_r, r \leq 2^s$, be all the gates in the bottom level of the circuit C such that there are more than s negative literals in their input, then

$$\begin{aligned} Pr[\tau_1 \neq 1 \vee \dots \vee \tau_r \neq 1] &\leq Pr[\tau_1 \neq 1] + \dots + Pr[\tau_r \neq 1] \\ &\leq 2^s (1/2)^{s+1} \\ &= 1/2 \end{aligned}$$

Thus,

$$Pr[\tau_1 \equiv 1 \wedge \dots \wedge \tau_r \equiv 1] \geq 1/2$$

Therefore, there is an assignment that converts the circuit $C_k^{m,2}$ to the circuit $C_{k-1}^{m,m}$ and eliminates all gates in the bottom level of the circuit C in whose input there are more than s negative literals. Let the circuit obtained from C by this assignment be C' .

Now partition the input of the function $f_{k-1}^{m,m}$ into groups of m variables each such that each group corresponds to the inputs to a bottom level gate of the tree circuit $C_{k-1}^{m,m}$. Randomly pick half of the variables in each group and assign them 1. The circuit $C_{k-1}^{m,m}$ under such an assignment is converted to the circuit $C_{k-1}^{m,m/2}$ defining the function $f_{k-1}^{m,m/2}$.

Let σ be an OR gate in the bottom level of the circuit C' with more than s positive literals in its input, then by Lemma 3.2,

$$Pr[\sigma \neq 1] \leq (1/2)^{s+1}$$

Let $\sigma_1, \dots, \sigma_t, t \leq 2^s$, be all the gates in the bottom level of the circuit C' with more than s positive literals in their input, then

$$Pr[\sigma_1 \neq 1 \vee \dots \vee \sigma_t \neq 1]$$

$$\begin{aligned} &\leq Pr[\sigma_1 \neq 1] + \dots + Pr[\sigma_t \neq 1] \\ &\leq 2^s (1/2)^{s+1} \\ &= 1/2 \end{aligned}$$

Thus,

$$Pr[\sigma_1 \equiv 1 \wedge \dots \wedge \sigma_t \equiv 1] \geq 1/2$$

Therefore, there is an assignment that converts the circuit $C_{k-1}^{m,m}$ to the circuit $C_{k-1}^{m,m/2}$ and eliminates all gates in the bottom level of C' that have more than s positive literals in their input. Let the circuit obtained from C' by this assignment be C'' .

Since each gate in the bottom level of the circuit C'' has neither more than s negative literals nor more than s positive literals in its input, the bottom fan-in of the circuit C'' is at most $2s = \frac{1}{6\sqrt{2(k-1)}}\sqrt{\frac{m}{\log m}}$, which is smaller than $\frac{m/2}{25e(k-1)\log m}$ for sufficiently large m . Thus, we have constructed a circuit C'' of depth $k - 1$, bottom fan-in less than $\frac{m/2}{25e(k-1)\log m}$, and size bounded by $2^s = 2^{\frac{1}{12\sqrt{2(k-1)}}\sqrt{\frac{m}{\log m}}}$ such that C'' computes the function $f_{k-1}^{m,m/2}$. This contradicts Theorem 2.2. \square

The following corollary will be used in Section 5.

Corollary 3.4 *The function $f_k^{m,2}$ can be computed by a circuit of depth k , linear size, and bottom fan-in 2, but cannot be computed by any depth $k - 1$ circuit of polynomial size.*

4 Trade-off between bottom fan-in and size

We first summarize the results in the previous two sections in the following theorem.

Theorem 4.1 *For all integers $b \geq 2$ and sufficiently large m , the function $f_k^{m,b}$ can be computed by a depth k circuit of linear size and bottom fan-in b , but requires size larger than $2^{\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}}$ for depth k circuits of bottom fan-in $\frac{b}{25ek \log m}$.*

PROOF. For the case $2 \leq b \leq k \log m$, since $\frac{b}{25ek \log m} < 1$, the theorem is implied by Theorem 3.3. The case $k \log m < b \leq \sqrt{km \log m/2}$ is proved in Theorem 2.1. For the case $\sqrt{km \log m/2} < b < 2\sqrt{km \log m/2}$, since $\frac{b}{25ek \log m} \leq \frac{\sqrt{km \log m/2}}{12k \log m}$, the theorem is implied by Theorem 2.1. Finally, the case $b \geq 2\sqrt{km \log m/2}$ is proved by Theorem 2.2. \square

A number of important consequences follow directly from Theorem 4.1.

Theorem 4.2 *For any integers $k \geq 1$ and $h \geq 1$, and for any real number r , there are functions that are computable*

by a circuit of linear size and depth k with bottom fan-in $O(\log^h n)$, but requires exponential size for depth k circuits of bottom fan-in $r \log^{h-1} n$.

By more careful selections of the bottom fan-in b in Theorem 4.1, combined with a padding technique, we are able to obtain general results for the trade-off between circuit size and circuit bottom fan-in. We illustrate this technique by the following theorem, which can be easily extended to other cases using the same technique.

Theorem 4.3 *For any integer $k \geq 1$ and for any real number $\epsilon > 0$, there is a function F_k^ϵ that is computable by a circuit of linear size and depth k with bottom fan-in $\log n$, but requires superpolynomial size for depth k circuits of bottom fan-in $O(\log^{1-\epsilon} n)$.*

PROOF. Choose h such that $\frac{h}{h+1} > 1 - \epsilon$, and then use Theorem 4.2 to choose a function $f_k^{m,b}$ of $\leq m^{k-1}$ variables which can be computed by a depth k circuit of linear size and bottom fan-in $b = 25ek \log^{h+1} m$ but requires size

$$2^{\frac{1}{12\sqrt{2k}} \sqrt{\frac{m}{\log m}}} \quad (1)$$

when the bottom fan-in is $\leq \log^h m$.

Now make the function $f_k^{m,b}$ formally the function F_k^ϵ of $n = 2^{25ek \log^{h+1} m}$ variables by adding dummy variables that are not used. The theorem now follows for the function F_k^ϵ since the size bound (1) is superpolynomial in n and $c \log^{1-\epsilon} n < \log^h m$ for any fixed constant c when m is sufficiently large. \square

In particular, if we let $\epsilon = 1$ and $h = 1$, then we obtain the following corollary that will be used in Section 5.

Corollary 4.4 *For any integer $k \geq 1$, there is a function F_k that is computable by a circuit of linear size and depth k with bottom fan-in $\log n$, but requires superpolynomial size for depth k circuits of bottom fan-in $O(1)$.*

5 Input read-modes of Turing machines

An important application of the above investigation is on the input read-modes of a sublinear-time alternating Turing machine, which is an important computational model in the study of complexity classes.

To make sublinear-time Turing machines meaningful, we allow a Turing machine to have a *random access input tape* plus a *read-write input address tape*, such that the Turing machine has access to the bit of the input tape denoted by the contents of the input address tape.

A number of input read-modes for sublinear-time alternating Turing machines have appeared in the literature. The standard input read-mode introduced by Chandra,

Kozen, and Stockmeyer [8] allows a computation path of an $O(\log n)$ -time alternating Turing machine to read up to $\Theta(\log n)$ input bits. Ruzzo [15] proposed an input read-mode in which each computation path can read at most one input bit and the reading must be performed at the end of the path. An input read-mode studied by Sipser [16] insists that the input address tape be always reset to blank after each input reading so that each input reading takes time $\Omega(\log n)$.

It can be shown that many complexity classes such as NC^k for $k \geq 1$ and AC^k for $k \geq 0$ remain the same for all these input read-modes of alternating Turing machines. On the other hand, it was unknown whether these input read-modes affect the classes of lower complexity such as the levels in the logarithmic time hierarchy. Recently, Cai and Chen [5] have given precise circuit characterizations for each level of the logarithmic time hierarchy based on each of the above three input read-modes. Combining these characterizations with the separation results given in the previous sections, we are able to show that all these input read-modes are distinct.

Formally, the *logarithmic time hierarchy* is defined to be the union of the following classes:

$$\Pi_1, \Pi_2, \dots, \Pi_k, \dots$$

where Π_k is the class of languages accepted by a log-time ATM that always starts with an \wedge -state and makes at most k alternations.

The above definition ignores the input read-modes of the log-time ATMs thus is not very precise. To be more precise, we will call

$$\Pi_1^U, \Pi_2^U, \dots, \Pi_k^U, \dots$$

the logarithmic time hierarchy *based on Chandra-Kozen-Stockmeyer's model*,

$$\Pi_1^R, \Pi_2^R, \dots, \Pi_k^R, \dots$$

the logarithmic time hierarchy *based on Ruzzo's model*, and

$$\Pi_1^S, \Pi_2^S, \dots, \Pi_k^S, \dots$$

the logarithmic time hierarchy *based on Sipser's model*, where Π_k^U (resp. Π_k^R, Π_k^S) is the class of languages accepted by a log-time ATM based on Chandra-Kozen-Stockmeyer's input read-mode (resp. on Ruzzo's input read-mode, on Sipser's input read-mode) that always starts with an \wedge -state and makes at most k alternations.

Theorem 5.1 ([5]) *For all integers $k \geq 1$,*

- (1) *If a language L is in the class Π_k^R , then L is accepted by a Π_k^{poly} -family of circuits;*
- (2) *If a language L is in the class Π_k^S , then L is accepted by a $\Pi_k^{\text{poly},c}$ -family of circuits for some constant c ;*
- (3) *If a language L is in the class Π_k^U , then L is accepted by a $\Pi_k^{\text{poly},d \log n}$ -family of circuits for some constant d ;*

Combining Theorem 1.1, Corollary 3.4, Corollary 4.4, and Theorem 5.1, we obtain the following strong separation of the logarithmic time hierarchy. The proof is omitted.

Theorem 5.2 For all $k \geq 1$, we have $\Pi_k^R \subset \Pi_k^S \subset \Pi_k^U \subset \Pi_{k+1}^R$, where \subset means “proper subset”.

Corollary 5.3 For all $k \geq 1$, the k th levels of the logarithmic time hierarchy based on Chandra-Kozen-Stockmeyer’s input read-mode, Sipser’s input read-mode, and Ruzzo’s input read-mode are all distinct.

References

- [1] M. AJTAI, Σ_1^1 -formulae on finite structures, *Ann. Pure Appl. Logic* 24, (1983), pp. 1-48.
- [2] D. BARRINGTON, N. IMMERMANN, AND H. STRAUBING, On uniformity within NC^1 , *J. Comput. System Sci.* 41, (1990), pp. 274-306.
- [3] R. B. BOPPANA AND M. SIPSER, The complexity of finite functions, in J. van Leeuwen, ed., *Handbook of Theoretical Computer Science Vol. A*, Elsevier, Amsterdam, 1990, pp. 757-804.
- [4] S. R. BUSS, The Boolean formula value problem is in ALOGTIME, *Proc. 19th Annual ACM Symposium on Theory of Computing*, (1987), pp. 123-131.
- [5] L. CAI AND J. CHEN, On input read-modes of alternating Turing machines, *Theoretical Computer Science* 148, (1995), pp. 33-55.
- [6] L. CAI AND J. CHEN, On the amount of nondeterminism and the power of verifying, *Lecture Notes in Computer Science 711 (MFCS'93)*, (1993), pp. 311-320. Journal version to appear in *SIAM Journal on Computing*.
- [7] L. CAI, J. CHEN, R. G. DOWNEY, AND M. R. FELLOWS, On the structure of parameterized problems in NP, *Information and Computation* 123, (1995), pp. 38-49.
- [8] A. K. CHANDRA, D. C. KOZEN, AND L. J. STOCKMEYER, Alternation, *J. Assoc. Comput. Mach.* 28, (1981), pp. 114-133.
- [9] J. CHEN, Characterizing parallel hierarchies by reducibilities, *Information Processing Letters* 39, (1991), pp. 303-307.
- [10] S. COOK, A taxonomy of problems with fast parallel algorithms, *Information and Control* 64, (1985), pp. 2-22.
- [11] M. FURST, B. SAXE, AND M. SIPSER, Parity, circuits, and the polynomial-time hierarchy, *Math. Systems Theory* 17, (1984), pp. 13-27.
- [12] R. L. GRAHAM, D. E. KNUTH, AND O. PATASHNIK, *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, Reading, MA, 1989.
- [13] J. HÅSTAD, *Computational limitations for small-depth circuits*, The MIT Press, Cambridge, MA, 1986.
- [14] J. HÅSTAD, Almost optimal lower bounds for small depth circuits, in S. Micali, ed., *Advances in Computing Research* 5, JAI Press Inc., Greenwich, 1989, pp. 143-170.
- [15] W. L. RUZZO, On uniform circuit complexity, *J. Comput. System Sci.* 22, (1981), pp. 365-383.
- [16] M. SIPSER, Borel sets and circuit complexity, *Proc. 15th Annual ACM Symposium on Theory of Computing*, (1983), pp. 61-69.
- [17] A. C. YAO, Separating the polynomial-time hierarchy by oracles, *Proc. 26th Annual IEEE Symposium on Foundations of Computer Science*, (1985), pp. 1-10.